LOVELY PROFESSIONAL UNIVERSITY

MAJOR PROJECT

Name: Mohammad Naseem

Registration No.:12317939

Topic: Image Blur/Sharpen with 2D Filters

Description: Applies convolution filters (blur or sharpen) to grayscale images using a C++ CGI backend and a user-friendly HTML/CSS frontend.

Live Project link:
https://github.com/MOhammadnaseem8329/Minor-Project/blob/main/Pro.cpp

Code :-
CPP

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <cstdlib>
using namespace std;

void parseContentType() {
   string s;
   getline(cin, s); // Skip boundary
}

void saveUploadedFile(const string& filename) {
   string line;
   ofstream out(filename, ios::binary);

   // Skip headers (until blank line)
   int blankCount = 0;
   while (getline(cin, line)) {
      if (line == "\r" || line.empty()) {
         blankCount++;
         if (blankCount == 2) break;
      }
   }

   // Read actual file content
   while (getline(cin, line)) {
      if (line.find("------WebKitFormBoundary") != string::npos) break;
      out << line << "\n";
   }

   out.close();
}

void readPGM(const string& filename, vector<vector<int>>& image, int& width, int& height, int&
maxVal) {
   ifstream file(filename);
   string magic;
   file >> magic >> width >> height >> maxVal;
   image.resize(height, vector<int>(width));
   for (int i = 0; i < height; ++i)
      for (int j = 0; j < width; ++j)
         file >> image[i][j];
   file.close();
}

void writePGM(const string& filename, const vector<vector<int>>& image, int width, int height, int
maxVal) {
   ofstream file(filename);
   file << "P2\n" << width << " " << height << "\n" << maxVal << "\n";
   for (const auto& row : image) {
      for (int val : row)
```

```
            file << val << " ";
        file << "\n";
    }
}

void applyFilter(const vector<vector<int>>& input, vector<vector<int>>& output,
                 const vector<vector<int>>& kernel, int divisor) {
    int h = input.size(), w = input[0].size();
    output = input;

    for (int i = 1; i < h - 1; ++i) {
        for (int j = 1; j < w - 1; ++j) {
            int sum = 0;
            for (int ki = -1; ki <= 1; ++ki)
                for (int kj = -1; kj <= 1; ++kj)
                    sum += input[i + ki][j + kj] * kernel[ki + 1][kj + 1];
            output[i][j] = max(0, min(255, sum / divisor));
        }
    }
}

int main() {
    cout << "Content-Type: text/html\n\n";

    char* contentLengthStr = getenv("CONTENT_LENGTH");
    if (!contentLengthStr) {
        cout << "<h2>Error: No content length</h2>";
        return 1;
    }

    int contentLength = atoi(contentLengthStr);
    cin.ignore(); // Skip line

    // Save uploaded file
    saveUploadedFile("input.pgm");

    // Read filter type from stdin again
    string postData;
    getline(cin, postData); // Contains `filter=blur` or `filter=sharpen`

    string filterType = postData.find("sharpen") != string::npos ? "sharpen" : "blur";

    // Process image
    int width, height, maxVal;
    vector<vector<int>> image, output;
    readPGM("input.pgm", image, width, height, maxVal);

    vector<vector<int>> blurKernel = {
        {1, 1, 1},
        {1, 1, 1},
        {1, 1, 1}
    };

    vector<vector<int>> sharpenKernel = {
        { 0, -1,  0},
        {-1,  5, -1},
        { 0, -1,  0}
    };
    if (filterType == "blur")
        applyFilter(image, output, blurKernel, 9);
    else
        applyFilter(image, output, sharpenKernel, 1);

    writePGM("output.pgm", output, width, height, maxVal);

    // Response
    cout << "<h2>Filter applied: " << filterType << "</h2>";
    cout << "<p><a href='/output.pgm' target='_blank'>Download output.pgm</a></p>";

    return 0;
}
```

HTML:

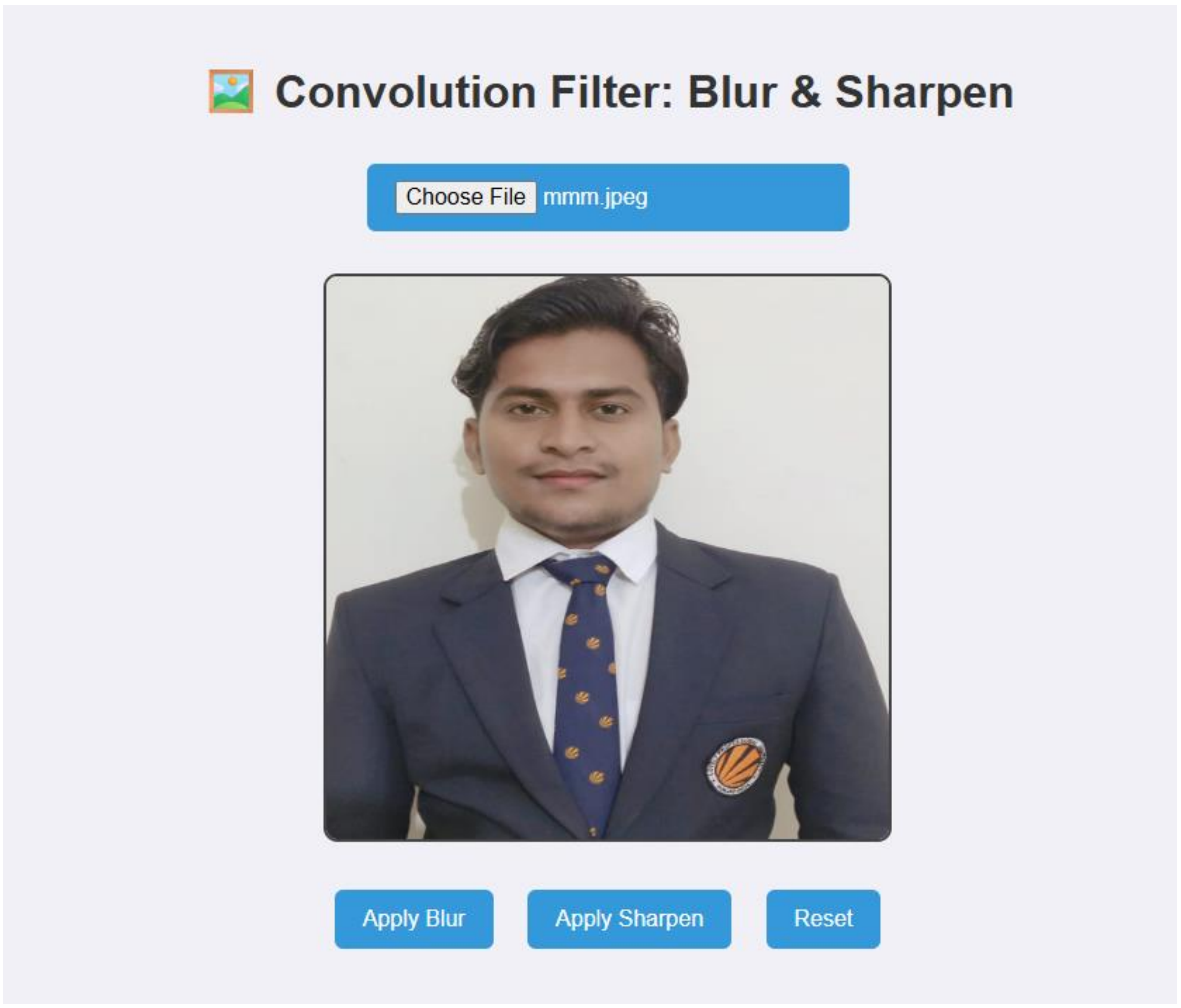
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Image Blur & Sharpen using Convolution</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js" defer></script>
</head>
<body>

  <h1>🖼️ Convolution Filter: Blur & Sharpen</h1>

  <input type="file" id="upload" accept="image/*">
  <br>
  <canvas id="canvas" width="400" height="400"></canvas>
  <br>

  <button onclick="applyFilter('blur')">Apply Blur</button>
  <button onclick="applyFilter('sharpen')">Apply Sharpen</button>
  <button onclick="reset()">Reset</button>

</body>
</html>
```

CSS:

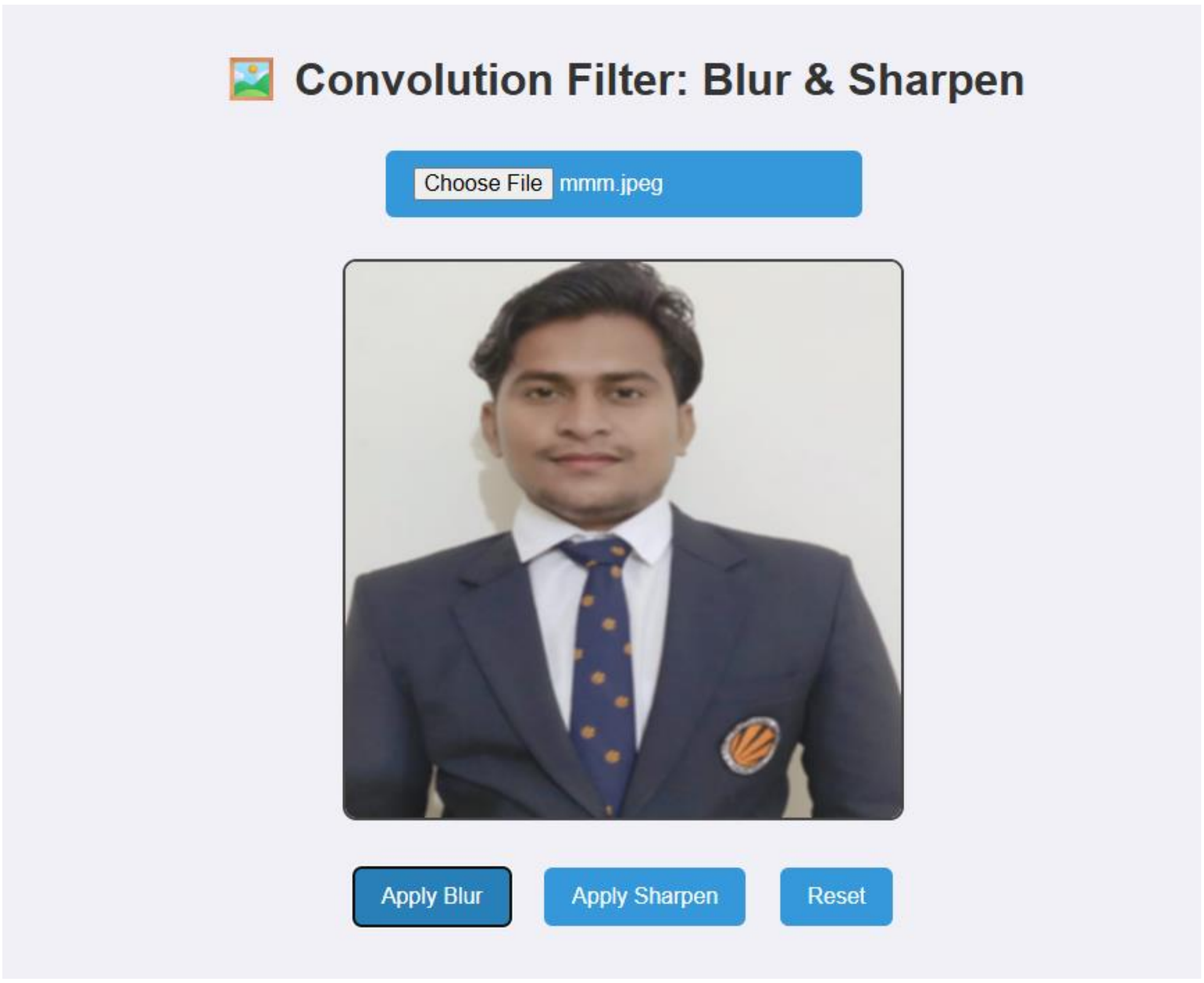
```
body {
  font-family: Arial, sans-serif;
  background: #f0f0f5;
  text-align: center;
  padding: 30px;
}

h1 {
  color: #333;
}

canvas {
  border: 2px solid #444;
  margin: 20px;
  border-radius: 10px;
}

button, input[type="file"] {
  margin: 10px;
  padding: 12px 20px;
  font-size: 16px;
  border-radius: 6px;
  border: none;
  background: #3498db;
  color: white;
  cursor: pointer;
}

button:hover {
  background: #2980b9;
}
```

Output :-

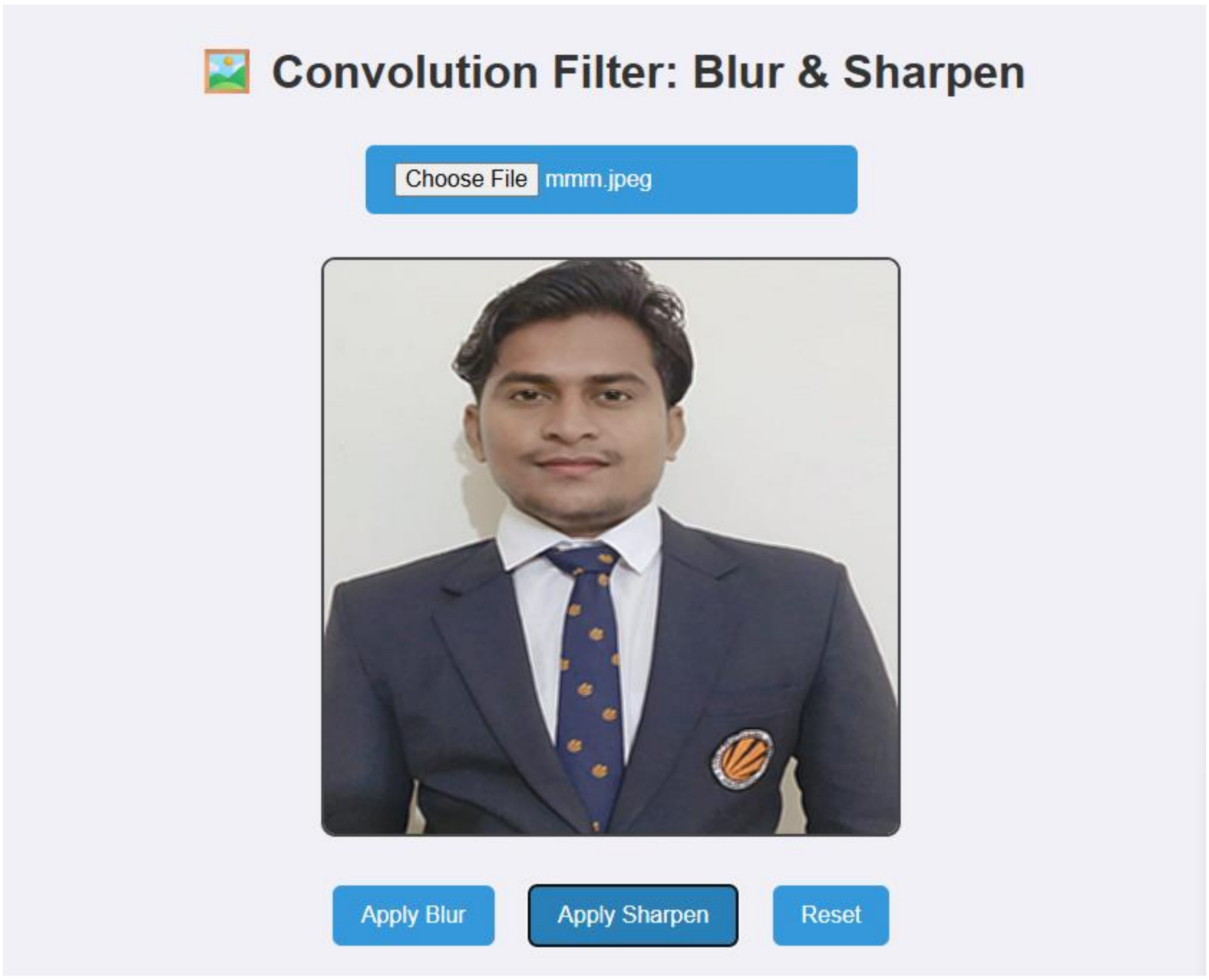
Normal image:



Blur image:


Convolution Filter: Blur & Sharpen
Choose File mmm.jpeg
Apply Blur
Apply Sharpen
Reset

Sharpen image:


🖼️ Convolution Filter: Blur & Sharpen
Choose File mmm.jpeg
Apply Blur
Apply Sharpen
Reset